LONG ISLAND
99ER USERS GROUP

VOL #12 NO.06 JUNE, 1993 $2.00

NO MEETING JUN (TONY'S HOME IS ALL PACKED UP TO MOVE.)

LONG ISLAND SOUND

EDITOR: FRANCIS J. BUBENIK JR.

PAGE 1.....TABLE OF CONTENTS.

PAGE 2.....1993 COMPUTOR FAIR SCHEDULE.
COMPILED BY FRANK J. BUBENIK JR.

PAGE 3.....XB MISCELLANEOUS #24.
BY EARL RAGUSE / LGOC U.G.

PAGE 4.....XB MISCELLANEOUS #24 (CONTINUED).

PAGE 5.....TIPS #71 TIGERCUB.
By JIM PETERSON.

PAGE 6.....TIPS #71 TIGERCUB (CONTINUED).
TIGERCUB SOFTWARE.

PAGE 7.....PROGRAMS THAT WRITE PROGRAMS #6.
By JIM PETERSON - TIGERCUB.

PAGE 8.....PROGRAMS THAT WRITE PROGRAMS #5.
CORCOMP AND MYARC REPAIR FACILITY.

PAGE 9.....TRANSFERRING FILES TI TO PC.
By PHILIP HARRIS - OTTAWA U.G.

PAGE 10.....MIKey - DESIGN NEWS. MAILER / LOGO

HAPPY FATHER'S DAY

ESTABLISHED APRIL-1983

COME TO THE TI FAIRS.

* 1993 FAIR SCHEDULE *

Compiled by Frank J. Bubenik Jr. (NL Editor)

JUN 12, 1993, (SAT) EDISON, NJ. RARITAN CENTER EXPO HALL. EXIT #10 NJ TNPK. 1200 TABLES. 10-3PM KGP.

JUL 31, 1993, (SAT) LI SHOW. HUNTINGTON HILTON HOTEL MELVILLE,NY EXIT#49-SOUTH RTE#110. 10-3PM 200 TABLES.

AUG 14, 1993 (SAT) FAIRLIEIGH DICKINSON UNIV. RTE 4 ON HACHENSACK AVE. HACKENSCK,N.J 10-3PM 500 TABLES.

AUG 28, 1993 (SAT) EDISON NJ. RARITAN CENTER EXP HALL 1200 TABLES. 10-3PM. EXIT#10 N.J TNPK. KGP.

TI NOV 12/13, 1993 (FRI/SAT) *11TH ANNUAL CHICAGO INTERNATIONAL FAIRE*. ELK GROVE HOLIDAY INN.HOTEL RES. 708) 437-6010 CODE "IWF". JIM DEARDS NEW CHAIRPERSON.

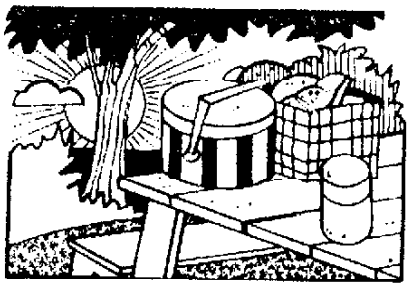
TI NOV 14, 1993 (SUN) MILWAUKEE TI FAIRE. INFO CALL GENE HITZ (414) 535-0133.

LITI 99ERS NEWSLETTER IS NOT RESPONCIBLE FOR CANCEL-ATIONS. CALL THE NUMBERS BELOW TO VERIFY TIME AND DATES BEFORE YOU GO.

* Ken Gordon Productions (KGP) SHOWS COST $8.00-$1.00 discount cards are sent to those people on there mailing lists. Call (800)631-0062 OR (908)297-2526 for info. (rev 2/20/93).

* Tri-State Computer Fairs (TSCF) SHOWS COST $6.00-$1.00 discount cards available by mail. Call Robert Barlow (201) 533-1991 for info. (rev 3/18/93).

Don't miss the bargains on computers, software, IC's, peripherals, printers, monitors, parts, supplies and books.



XB MISCELLANEOUS #24
By Earl Raguse

ELEMEMTARY XB PROGRAMMING (CONT) THIS ARTICLE IS PART 7 IN AN 8 PART SERIES THAT STARTED WITH XB MISCELLANY #18. IF YOU ARE A NEW READER, AND DO NOT HAVE ACCESS TO REST OF THIS SERIES, PLEASE CONTACT ME OR THE NEWSLETTER EDITOR.

WELL, I THINK I HAVE FINALLY MADE IT, AT LAST TO THE PRINT MODULE. I HOPE I HAVEN'T GIVEN YOU THE IDEA THAT THIS WAS ALL JUST A COUPLE OF EVENINGS WORK FOR ME. MAYBE IT WAS TO GET THE FIRST CODE WRITTEN, BUT THEN STARTS THE TESTING. I AM VERY ERROR PRONE, SOMETIMES I EVEN APPOACH STUPIDITY. BUT IN THE END, GETTING A PROGRAM TO WORK IS FAIRLY EASY.

NOW GETTING ONE TO WORK LIKE I WANT IT TO WORK IS SOMETHING ELSE AGAIN. IT STILL IS NOT PERFECT, AND I WILL TELL YOU ABOUT SOME RECOMMENDED CHANGES. ONCE A PROGRAM IS WORKING, SAY AFTER TWO DAYS, THEN I START IMPROVING. I FIDDLE WITH THE PROMPTS, THE LOCATION OF THE SCREEN DISPLAYS, DO PARTIAL SCREEN CLEARING, ETC ETC. THEN I START ADDING LITTLE FEATURES THAT SEEM NICE, BUT THEY ALWAYS GET ME INTO MORE TROUBLE THAN I BARGAINED FOR. LIKE ADDING FWD/REV, THEN OVERRIDING IT WITH PAR (PRINT ALL RECORDS). THEN I NEED A WAY TO HALT THE PROCESS.

THAT MAY GO ON FOR A WEEK, AND IN THIS CASE IT DID. I CHANGED THE WAY THE DISPLAY AND EDIT MODULES WORKED FOUR TIMES. I FINALLY ENDED UP USING THE SAME TECHNIQUE I WAS USING FOR EDIT. PRINT, NOW ALSO HAS SIMILAR FEATURES OF BEING ABLE TO FORAGE THROUGH THE RECORDS FWD AND REW, OR JUST DO ALL FROM EITHER DIRECTION FROM ANY PLACE. I ADDED HALT TO THE PRINT MODULE, BECAUSE I COULD THINK OF A COUPLE LONG LISTS THAT I HAVE THAT I WOULDN'T WANT TO GO THROUGH THE WHOLE THING BEFORE STOPPING, ONCE STARTED.

I WILL HAVE TO REMEMBER THAT, I WILL USE THAT AGAIN. ACTUALLY ITS NOT TO FAR FROM TIPSLABEL. THE PRINT MODULE USES THE SAME METHOD. IT EASIER HERE, BECAUSE THE ENTIRE FILE IS IN AN ARRAY, NOT READ FROM DISK A RECORD AT A TIME AS WITH TIPSLABEL. THERE I WAS AFRAID I DID NOT HAVE ENOUGH DATA MEMORY LEFT AFTER PROGRAM SPACE.

SO MUCH FOR DIGRESSIONS, LETS LOOK AT THE PRINT MODULE.

LINES 1600-10 DO OLD HAT THINGS. LINE 1620 CALLS UP A NEW SUBPROGRAM OPENPRNTR(P$,M), SEE LINE 6410, AND EMPHASIZE(M), LINE 6560.

THE NICE THING ABOUT OPENPRNTR IS THAT IT WARNS YOU TO "TURN ON YOUR PRINTER" IF YOU FORGET TO TURN IT ON. IF IT IS ON, THE MESSAGE ONLY FLASHES FOR A MOMENT, BUT IF YOU HAVEN'T IT WILL DISPLAY THE MESSAGE AND WAIT FOR YOU TO COMPLY. LINE 130 OF THE MAIN PROGRAM HAS A VARIABLE P$="PIO". IF THIS IS NOT THE NAME YOUR PRINTER ANSWERS TO, CHANGE IT THERE. IT HAS NO EFFECT ON OPENPRNTR DIRECTLY. IT EXPECTS THE PROPER NAME IN P$.

LINE 1620 ALSO SETS J,DIR=1 AND PAR=0.

LINES 1625-30 DISPLAY THE RECORD NUMBER, PARTIALLY CLEAR THE SCREEN, DISPLAY THE SELECTED RECORD VIA SUBROUTINE 2100, AND CHECKS IF PAR=1, IF SO THEN 1680 TO PRINT THE RECORD.

LINES 1640-50 PUT UP A PROMPT "NEXT LAST PRINT ALL QUIT", GO CHECKS ON THE SELECTION, AND ON DIRECTS THE BRANCHING. IF NEXT IS SELECTED, THEN WE GO TO LINE 1650 DIR=1. IF LAST IS SELECTED, THEN WE GO TO 1655, AND DIR=-1, EITHER LINE GOES TO 1660 WHERE J=J+DIR. THIS IS EQUIVALENT TO FWD/REV. THE VALUE OF J IS CHECKED AGAINST THE LIMITS OF 1 AND N, IF OK, GOTO 1625 AND START OVER, IF THE LIMITS OF J ARE EXCEEDED, IT GOES TO LINE 1685 TO CLOSE THE FILE AND DISPLAY "THAT'S ALL". THEN FALLS THROUGH TO CALL PAK AND RETURN TO MENU (140)

NOW IF ALL IS SELECTED THE PROGRAM GOES TO 1670, AND PAR=1, WHICH STARTS AUTOMATIC PRINTING OF ALL RECORDS. TO WARN OF THIS, THE PROGRAM PUTS A PROMPT TO WIT: "TO HALT, HOLD H KEY".

LINE 1680, IN ADDITION TO PRINTING THE RECORD, CHECKS CALL KEY TO SEE IF H IS PRESSED, IF SO THEN 1640, WHERE PAR=0 ELSE 1660 TO ADVANCE ANOTHER RECORD.

OF COURSE, IF QUIT IS SELECTED, WE WILL RETURN TO MENU, VIA 1690.

THAT COMPLETES THE PROGRAM AS IT CURRENTLY STANDS. I DO HAVE ONE SMALL ADDITON, WHICH IS SHOWN IN LINES 135 AND 136. THEY PROVIDE THE OPTION OF ESCAPING FROM THE PROGRAM ONCE YOU KNOW WHAT IT DOES. IT ALSO GIVES A LITTLE OPERATING HELP.

NOW, I SAID I HAD SOME RECOMMENDATIONS FOR CHANGE. WELL, I DID IT INSTEAD. ONE TASK WAS TO REWITE THE ALGORITHM FOR THE FILES SO AS NOT TO KEEP TRACK OF THE NUMBER OF RECORDS. THAT IS REALLY ONLY USED AS A LOOP LIMIT ANYWAY, THE ACTUAL NUMBER OF RECORDS READ IS WHAT IS RETURNED, AND USED FOR KEEPING RECORD ARRAYS IN ORDER. EOF IS USED TO DETERMINE WHEN ALL RECORDS ARE READ. TO DO THIS I HAD TO CHANGE SAVIT SO THAT IT DID NOT PRINT N TO THE FILE, ONLY USED IT AS THE NUMBER OF RECORDS TO PRINT. GETIT WAS GIVEN A FIXED LOOP LIMIT OF 200. THAT IS THE LIMIT ON DIM OF REC$(,) ANYWAY. THIS MAKES LISTMAN MORE LIKE MY EXISTING FILES. I HAD FULLY INTENDED TO LEAVE THAT SMALL DETAIL TO YOU, BUT I FOUND TIME SO I HAVE ALREADY DONE IT. LUCKY YOU. IF YOU NEED HELP, PLEASE CALL ME, 714/847-5875.

I PLAN TO DO SOME FOLLOW-UP ON THIS FILE TO ANSWER QUESTIONS YOU MUST HAVE BY NOW. BUT, YOU ARE GOING TO HAVE TO TELL ME WHAT THOSE QUESTIONS ARE. BE LIKE LOGO, WHEN I TELL YOU TO SQUIRT, AND YOU DO NOT KNOW HOW TO SQUIRT, YOU SHOULD SAY, "TELL ME HOW TO SQUIRT".

THIS IS THE PRINT MODULE

```
1600 CALL CLPUT("PRINT LABEL
S SECTION",2)
1610 IF N<1 THEN CALL PUT("Y
OU HAVE NO DATA TO PRINT",12
):: GOTO 1690
1620 M=4 :: CALL OPENPRNTR(P
$,M):: CALL EMPHASIZE(M):: J
,DIR=1 :: PAR=0
1625 DISPLAY AT(7,10):"RECOR
D ";J
1630 CALL CLS(24,24) :: GOSU
B 2100 :: IF PAR THEN 1680
1640 CALL PUT("NEXT LAST PRI
NT ALL QUIT",23):: PAR=0
1645 CALL GO("NLPAQ",Q):: IF
 Q=0 THEN 1645 ELSE ON Q GOT
O 1650,1655,1680,1670,1685
1650 DIR=1 :: GOTO 1660
1655 DIR=-1 :: GOTO 1660
1660 J=J+DIR :: IF J<1 OR J>
N THEN 1685 ELSE 1625
1670 PAR=1 :: CALL PUT("TO H
ALT, HOLD H KEY",5)! PRINT A
LL RECORDS
1680 PRINT #M:CHR$(10):: FOR
 K=1 TO 5 :: PRINT #M:REC$(J
,K):: NEXT K :: CALL KEY(3,K
,S):: IF K=ASC("H")THEN 1640
 ELSE 1660
1685 CLOSE #M :: CALL CLS(22
,23):: CALL PUT("THAT'S ALL"
,20)
1690 CALL PAK :: GOTO 140
```

TIPS FROM THE TIGERCUB

No. 71

Tigercub Software
156 Collingwood Ave.
Columbus, OH 43213

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to $5.00 each. I am out of printed documentation so it will be supplied on disk.

My TI-PD library now has over 600 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename. Basic programs converted to XBasic, etc. The price is just $1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #6 is available for $1 which is deductible from the first order.

Newsletters recently have been reprinting a random music player from several years ago. I thought I would try to make it a bit more musical -

```
100 REM ECHO2
110 RANDOMIZE
120 DATA 165,196,247,262,330,392,523
130 DATA 131,175,220,262,349,440,523
140 DATA 147,175,196,247,294,392,494
150 FOR J=0 TO 6 :: READ C(J):: NEXT J
160 FOR J=0 TO 6 :: READ F(J):: NEXT J
170 FOR J=0 TO 6 :: READ G(J):: NEXT J
180 X,Y,Z=INT(RND*7):: GOTO 200
190 Z=Y :: Y=X :: X=INT(RND*7)
200 T=T+1+(T=18)*18 :: IF T>12 THEN GOSUB 240 ELSE IF T>6 THEN GOSUB 230 ELSE GOSUB 220
210 CALL KEY(0,K,S):: IF S=0 THEN 190 ELSE STOP
220 CALL SOUND(-300,C(X),0,C(Y),9,C(Z),19):: RETURN
230 CALL SOUND(-300,F(X),0,F(Y),9,F(Z),19):: RETURN
240 CALL SOUND(-300,G(X),0,G(Y),9,G(Z),19):: RETURN
```

In a previous tips, I had a short routine to compute First Class postage. In the Bluegrass 99ers newsletter, Mark Schafer published an even shorter version -

```
100 INPUT "OUNCES? ":A :: PRINT .06-.23*INT(-A):: GOTO 100
```

Mark gives some other useful algorithms. INT(X) will round DOWN to the nearest integer, but remember that a negative number is also rounded downward, therefore INT(-4.1)=5.

INT(X+.5) rounds a number to the NEAREST integer; a number ending on .5 is rounded up.

-INT(-X) rounds UPWARD to the next integer, so that -INT(-4.1)=5

-INT(.5-X) rounds to the NEAREST integer but a number ending in .5 is rounded down instead of up.

INT(X*10+.5)/10 rounds to the nearest tenth.

INT(X*10)/10 truncates to the nearest tenth.

Thanks, Mark

I still like to program "brain games", so here is another one. I developed it from a much simpler version in an ad for membership in the Mensa Society of really brainy people, so I call it the Super Mensa Puzzle. It's not so hard at level 1 or 2.

```
100 CALL CLEAR :: FOR S=0 TO 12 :: CALL COLOR(S,2,16):: NEXT S :: CALL SCREEN(5)
110 DISPLAY AT(3,3)ERASE ALL :"THE SUPER MENSA PUZZLE":"" :" X restart  ? help  0 quit":"":"  P +  M -  T *  D /"
120 !by Jim Peterson
130 DISPLAY AT(9,1):" Put a P(lus) or +, or":"M(inus) or -, or T(imes) or":"*, or D(ivide by) or / in"
140 DISPLAY AT(12,1):"each blank to reach the":"total on the sprite."
150 DISPLAY AT(15,1):" Use X to start over, ? to":"give up and see answer, 0 to":"quit."
160 DISPLAY AT(19,1):"Difficulty level (1 - 5)?" :: CALL CALLKEY(19,27,"12345",L$):: L=VAL(L$):: DISPLAY AT(19,1):RPT$(" ",254)
170 DISPLAY AT(19,1):""
180 FOR J=0 TO 4+L :: Z$=Z$&STR$(J)&" " :: NEXT J :: Z$=SEG$(Z$,1,LEN(Z$)-1)&"=" :: ZC=(4+L)*2+3
190 RANDOMIZE :: GOSUB 320
200 DISPLAY AT(12,1):Z$&STR$(T@):"":"":"" :: C=2 :: N=0 :: N2=1 :: T=0 :: V$="+P-M*T/DX?!@" :: CALL DELSPRITE(ALL)
210 CALL MAGNIFY(2):: S$=STR$(T@):: R=63 :: CC=95 :: FOR J=1 TO LEN(S$):: CALL SPRITE(#J,ASC(SEG$(S$,J,1)),5,R,CC):: CC=CC+20 :: NEXT J
220 CALL CALLKEY(12,C,V$,X$)
230 ON POS(V$,X$,1)GOSUB 280,280,290,290,300,300,310,310,200,370,370,390
240 IF C<ZC-2 THEN 220 ELSE IF T=T@ THEN 260
250 FOR J=1 TO 20 :: DISPLAY AT(15,6):"WRONG!" :: DISPLAY AT(15,6):"wrong!" :: NEXT J :: DISPLAY AT(15,6):"" :: GOTO 200
260 IF X$="I" OR X$="?" THEN 270 ELSE DISPLAY AT(15,6):"RIGHT!" :: CALL SOUND(200,196,5):: CALL SOUND(500,523,2) :: GOSUB 320 :: GOTO 200
270 DISPLAY AT(15,5):"PRESS ANY KEY" :: DISPLAY AT(15,5):"press any key" :: CALL KEY(0,K,S):: IF S=0 THEN 270 ELSE GOSUB 320 :: GOTO 200
280 DISPLAY AT(12,C):"+":: T=T+N2 :: GOSUB 380 :: RETURN
290 DISPLAY AT(12,C):"-":: T=T-N2 :: GOSUB 380 :: RETURN
300 DISPLAY AT(12,C):"*":: T=T*N2 :: GOSUB 380 :: RETURN
310 DISPLAY AT(12,C):"/":: T=T/N2 :: GOSUB 380 :: RETURN
320 T=0 :: FOR J=1 TO 4+L
330 X=INT(RND*4+1):: IF X=1 THEN T=T+J :: P$(J)="+" :: GOTO 350 ELSE IF X=2 THEN T=T-J :: P$(J)="-" :: GOTO 350 ELSE IF X=3 THEN T=T*J :: P$(J)="*" :: GOTO 350
340 IF T/J<>INT(T/J)THEN 330 ELSE T=T/J :: P$(J)="/"
350 NEXT J
360 IF T>9999 OR T<-999 THEN 320 ELSE T@=T :: RETURN
370 DC=2 :: FOR J=1 TO 4+L :: DISPLAY AT(12,DC):P$(J)::: DC=DC+2 :: ON POS(V$,P$(J),1)GOSUB 280,280,290,290,300,300,310,310 :: NEXT J
380 T$=STR$(T):: T$=SEG$(T$,1,13-L):: DISPLAY AT(12,ZC):T$ :: C=C+2 :: N=N+1 :: N2=N2+1 :: RETURN
390 CALL CLEAR :: STOP
400 SUB CALLKEY(R,C,V$,K$)
410 CALL HCHAR(R,C+2,30):: FOR T=1 TO 3 :: CALL KEY(3,K,S):: IF S<>0 THEN 440
420 NEXT T :: CALL HCHAR(R,C+2,20):: FOR T=1 TO 3 :: CALL KEY(3,K,S):: IF S<>0 THEN 440
430 NEXT T :: GOTO 410
440 IF POS(V$,CHR$(K),1)=0 THEN 410 ELSE K$=CHR$(K)
450 SUBEND
```

Programmers might be interested in the CALLKEY subprogram in lines 400-450 which allows a one-character ACCEPT without pressing Enter, with blinking cursor and with validation. Since the - and * and + symbols require the Shift key, I provided alternative P, M, T and D key input - in either upper or lower case, since key mode 3 accepts either as upper case.

The Line newsletter has a tip by Andy Frueh on converting a DV40 file to DV80 by using a sector editor to find find byte >11 of the

NEXT PAGE

header sector and change it from >28 to >50. If you dislike sector editing and hex numbers as much as I do, you might prefer this method. It comes in handy to convert program listings, listed as DV28 files in 28-column size with Super Extended Basic, into DV80 files so that I can get them into Funnelweb to incorporate them into my text, such as this -

```
100 DISPLAY AT(12,1)ERASE AL
L:"Input file? DSK" :: ACCEP
T AT(12,16):IN$ :: DISPLAY A
T(14,1):"Record length?" ::
ACCEPT AT(14,16):IL
110 DISPLAY AT(16,1):"Output
 file? DSK" :: ACCEPT AT(16,
17):OUT$ :: DISPLAY AT(18,1)
:"Record length?" :: ACCEPT
AT(18,16):OL
120 OPEN #1:"DSK"&IN$,VARIAB
LE IL :: OPEN #2:"DSK"&OUT$,
VARIABLE OL
130 LINPUT #1:M$ :: PRINT #2
:M$ :: IF EOF(1)<>1 THEN 130
 ELSE CLOSE #1 :: CLOSE #2
```

After I published an article about writing a program to add blanks to records in a file, Stephen Shaw wrote to me from England and Chas. Stringer wrote from Decatur to point out that I could have done the same thing with TI-Writer or Funnelweb Replace String. I had never realized that you could specify columns with the RS command.

For instance, to get a left margin of two spaces, use CTRL O to get the open cursor, put the cursor at the beginning of the first line. use FCTN 8 to put a blank first line above it, then FCTN 9 and RS and enter 0 0 // / and then A for all lines.

Stephen Shaw published in MICROpendium a routine to add fractions, and challenged anyone to improve it to reduce the result. Dean Mah sent in a subprogram to do the reduction. Bruce Harrison sent me this greatly speeded-up version.

```
1 ! ADD FRACTIONS WITH REDUC
TION
2 ! MODIFIED FOR REDUCTION A
ND SPEED ENHANCEMENT BY BRUC
E HARRISON
3 ! ORIGINAL PROGRAM BY STEP
HEN SHAW
4 ! VERSION ADFR6 OF 1 APRIL
 93
90 ON WARNING NEXT
100 DISPLAY AT(10,5)ERASE AL
L:"--- + --- = ---"
110 ACCEPT AT(9,5)SIZE(3)VAL
IDATE(DIGIT):A
120 ACCEPT AT(11,5)SIZE(3)VA
LIDATE(DIGIT):B :: IF B=0 TH
EN 120
130 ACCEPT AT(9,11)SIZE(3)VA
LIDATE(DIGIT):C
140 ACCEPT AT(11,11)SIZE(3)V
ALIDATE(DIGIT):D :: IF D=0 T
HEN 140
150 FOR L=MAX(B,D)TO B*D STE
P MAX(B,D)
160 IF INT(L/B)<L/B THEN 180
170 IF INT(L/D)=L/D THEN 190
180 NEXT L
190 N=L/B*A+L/D*C
200 FOR Y=2 TO MIN(N,L)
210 IF N/Y>INT(N/Y)THEN 230
220 IF L/Y=INT(L/Y)THEN N=N/
Y :: L=L/Y :: GOTO 200
230 NEXT Y
240 DISPLAY AT(9,16):USING "
####":N
250 DISPLAY AT(11,16):USING
"####":L
260 DISPLAY AT(14,3):"PRESS
ENTER FOR ANOTHER"
270 DISPLAY AT(1,1):"NORMAL
REBULT=";A/B+C/D
280 CALL KEY(0,K,S):: IF S<>
1 THEN 280 ELSE IF K=13 THEN
 100
```

This one was published long ago in Tips #31 but someone asked for an encoding and program. This one will make codes that should be quite difficult to crack. Both parties must have the program with the same random code created by CODEPRINT merged in, but you can make any number of merge files and merge them in according to the day of the week, etc.

First we need one of those those programs that write a program -

```
100 !CODEPRINT by Jim Peters
on - creates a random code i
n a MERGE format program COD
ESTRING to be MERGEd into CO
DEMAKER
110 FOR J=1 TO 254 :: M$=M$&
CHR$(J):: NEXT J
120 FOR J=1 TO 254 :: RANDOM
IZE :: X=INT(RND*LEN(M$)+1):
: C$=C$&SEG$(M$,X,1):: M$=SE
G$(M$,1,X-1)&SEG$(M$,X+1,LEN
(M$)):: NEXT J
130 OPEN #1:"DSK1.CODESTRING
",VARIABLE 163,OUTPUT :: PRI
NT #1:CHR$(0)&CHR$(1)&"C$"&C
HR$(190)&CHR$(199)&CHR$(127)
&SEG$(C$,1,127)&CHR$(0)
140 PRINT #1:CHR$(0)&CHR$(2)
&"C2$"&CHR$(190)&CHR$(199)&C
HR$(127)&SEG$(C$,128,127)&CH
R$(0)
150 PRINT #1:CHR$(0)&CHR$(3)
&"C$"&CHR$(190)&"C$"&CHR$(18
4)&"C2$"&CHR$(0):: PRINT #1:
CHR$(255)&CHR$(255):: CLOSE
#1 :: END
```

And now the coder/decoder -

```
100 !TIGERCUB CODEMAKER writ
ten by Jim Peterson
110 !The MERGE format progra
m CODESTRING created by the
program CODEPRINT must be ME
RGEd into lines 1-3 of this
program
120 DIM A$(254):: DISPLAY AT
(3,6)ERASE ALL:"TIGERCUB COD
EMAKER" :: DISPLAY AT(12,1):
"Do you want to": :"(1)Encod
e":"(2)Decode"
130 CALL KEY(0,K,ST):: IF K=
49 THEN 140 ELSE IF K=50 THE
N 290 ELSE 130
140 OPEN #1:"DSK1.CODE",VARI
ABLE 254,OUTPUT
150 DISPLAY AT(5,6)ERASE ALL
:"Type message in segments o
f":"not more than 254 charac
ters":"and Enter. When done,
 type"
160 DISPLAY AT(9,1):"END and
 Enter. Type slowly":"to avo
id skipped characters.":"Bac
kspace with FCTN S to":"corr
ect.": :"Press any key"
170 CALL KEY(0,K,ST):: IF ST
=0 THEN 170
180 CALL CLEAR :: CALL LONGA
CCEPT(0,M$):: IF M$="END" TH
EN 280
190 DISPLAY AT(20,1):"WAIT,
PLEASE - ENCODING"
200 FOR J=1 TO LEN(M$)
210 A$(ASC(SEG$(C$,J,1)))=SE
G$(M$,J,1)
220 NEXT J
230 FOR J=1 TO 254 :: RANDOM
IZE
240 IF A$(J)="" THEN A$(J)=C
HR$(INT(26*RND+65))
250 CODE$=CODE$&A$(J)
260 NEXT J :: PRINT CODE$
270 PRINT #1:CODE$ :: CODE$=
"" :: FOR J=1 TO 254 :: A$(J
)="" :: NEXT J :: GOTO 180
280 CLOSE #1 :: END
290 OPEN #1:"DSK1.CODE",VARI
ABLE 254,INPUT :: CALL CLEAR
 :: DISPLAY AT(12,10):"DECOD
ING"
300 LINPUT #1:CODE$ :: FOR J
=1 TO 254 :: M$=M$&SEG$(CODE
$,ASC(SEG$(C$,J,1)),1):: NEX
T J :: PRINT M$;;; M$=""
310 IF EOF(1)<>1 THEN 300 ::
 CLOSE #1 :: END
320 SUB LONGACCEPT(L,M$):: X
=0 :: IF L<>0 THEN R=L ELSE
R=R+1
330 M$="" :: C=3 :: CH=140 :
: CALL CHAR(140,RPT$("0",14)
&"FF")
340 CALL HCHAR(R,C,CH):: CH=
CH+5+(CH=160)*25 :: CALL KEY
(0,K,ST):: IF ST<1 THEN 340
350 IF K<>8 THEN 370 :: X=X-
1 :: C=C-1 :: IF C=2 THEN C=
30 :: R=R-1
360 M$=SEG$(M$,1,LEN(M$)-1):
: GOTO 340
370 IF K=13 THEN 410
380 X=X+1 :: M$=M$&CHR$(K)::
 CALL HCHAR(R,C,K):: IF K=25
4 THEN 410
390 C=C+1 :: IF C=31 THEN C=
3 :: R=R+1 :: IF R=25 THEN C
ALL CLEAR :: R=1
400 GOTO 340
410 R=0 :: SUBEND
```

Memory full,
Jim Peterson

PROGRAMS THAT WRITE PROGRAMS
PART 6
by Jim Peterson

The first five parts of this series were written long ago, but since then I have found a new method to write programs that really do write programs. I must give Karl Ronstedt credit for this idea.

To illustrate this technique, I will use a program which writes an auto-loader to display a diskfull of programs by their complete name rather than the abbreviated filename. This is the LOAD program which I put on all my TI-PD disks.

First, we key in the part which will always be a part of the LOAD program. Do not change the line numbers because there is a reason for them, and leave that REM in line 11 because something else will be plugged in there later.

```
10 CALL CLEAR :: DIM M$(127) :: CALL SCREEN(5):: FOR S=0
 TO 14 :: CALL COLOR(S,2,8):: NEXT S :: CALL PEEK(8198,A
):: IF A<>170 THEN CALL INIT
11 REM
12 ON WARNING NEXT
13 X=X+1 :: READ M$(X):: IF M$(X)<>"END" THEN 13
14 R=3 :: FOR J=1 TO X-1 :: READ X$ :: DISPLAY AT(R,1):S
TR$(J);TAB(4);X$ :: R=R+1 :: IF R<23 THEN 17
15 DISPLAY AT(24,1):"Choice? or 0 to continue 0" :: ACCE
PT AT(24,26)VALIDATE(DIGIT)SIZE(-3):N :: IF N>X-1 THEN 1
5
16 IF N<>0 THEN 19 :: R=3
17 NEXT J
18 DISPLAY AT(24,1):"Choice?" :: ACCEPT AT(24,9)VALIDATE
(DIGIT):N :: IF N=0 OR N>X-1 THEN 18
19 CALL CHARSET :: CALL CLEAR :: CALL SCREEN(8):: CALL P
EEK(-31952,A,B):: CALL PEEK(A*256+B-65534,A,B):: C=A*256
+B-65534 :: A$="DSK1."&M$(N) :: CALL LOAD(C,LEN(A$))
20 FOR J=1 TO LEN(A$):: CALL LOAD(C+J,ASC(SEG$(A$,J,1)))
:: NEXT J :: CALL LOAD(C+J,0):: GOTO 10000
10000 RUN "DSK1.1234567890"
```

Now, save that "source code" by SAVE DSK1.CAT/S,MERGE. Then key in this "assembler" which will convert the "source code" into an "object code."

```
100 OPEN #1:"DSK1.CAT/S",VARIABLE 163,INPUT
110 OPEN #2:"DSK1.CAT/O",VARIABLE 163,OUTPUT
120 FOR J=10 TO 21 :: LINPUT #1:M$ :: PRINT #2:CHR$(0)&C
HR$(J)&CHR$(156)&CHR$(253)&CHR$(200)&CHR$(1)&"2"&CHR$(18
1)&CHR$(199)&CHR$(LEN(M$))&M$&CHR$(0):: NEXT J
130 PRINT #2:CHR$(255)&CHR$(255):: CLOSE #1 :: CLOSE #2
```

Note what this routine does. It reads in each line of the tokenized CAT/S and prints it back out to CAT/O preceded by line numbers 10 to 21 in tokenized two-byte format followed by the tokens for PRINT #2, the tokens for a quoted string followed by the CAT/S record and the CHR$(0) end-of-line indicator. Then it prints the double 255 end-of-file indicator and closes the files.

Now key in the CATWRITER program.

```
1 CALL CLEAR :: CALL TITLE(16,"CATWRITER"):: CALL CHAR(1
27,"3C4299A1A199423C"):: DISPLAY AT(2,10):"Version 1.4":
:TAB(8);" Tigercub Software"
2 DISPLAY AT(15,1):"For free":"distribution":"but no pri
ce or":"copying fee":"to be charged." :: FOR D=1 TO 500
:: NEXT D :: CALL DELSPRITE(ALL)
3 DISPLAY AT(2,3)ERASE ALL:"TIGERCUB CATWRITER V.1.4":::
"Will read a disk directory,":"request an actual progra
":"name for each program-type"
4 DISPLAY AT(7,1):"filename, and create a merg-":"able Q
uickloader which dis-":"plays full program names and":"r
uns a selected program."
5 DISPLAY AT(12,1):" Place disk to be cataloged";"in dri
ve 1 and press any key" :: CALL KEY(0,K,S):: IF S=0 THEN
 5
9 OPEN #2:"DSK1.CAT",VARIABLE 163,OUTPUT
100 OPEN #1:"DSK1.",INPUT,RELATIVE,INTERNAL :: INPUT #1:
N$,A,J,K :: LN=1000 :: FN=1100
110 DISPLAY AT(12,1):"Disk name?":::N$ :: ACCEPT AT(14,1
)SIZE(-28):N$ :: LX$=STR$(14-LEN(N$)/2):: LXLEN=LEN(LX$)
120 PR$=CHR$(0)&CHR$(11)&CHR$(162)&CHR$(248)&CHR$(183)&C
HR$(200)&CHR$(1)&"1"&CHR$(179)&CHR$(200)&CHR$(LXLEN)&LX$
130 PR$=PR$&CHR$(182)&CHR$(181)&CHR$(199)&CHR$(LEN(N$))&
N$&CHR$(0)
140 PRINT #2:PR$
145 DISPLAY AT(23,1):"To omit a file, press Enter"
150 X=X+1 :: INPUT #1:F$,A,J,B :: IF LEN(F$)=0 THEN 190
:: IF ABS(A)=5 OR ABS(A)=4 AND B=254 THEN 160 ELSE X=X-1
 :: GOTO 150
160 DISPLAY AT(12,1):F$;"PROGRAM NAME?" :: ACCEPT AT(14,
1)SIZE(25):F$ :: IF F$="" THEN X=X-1 :: GOTO 150
170 PRINT #2:CHR$(INT(FN/256))&CHR$(FN-256*INT(FN/256))&
CHR$(147)&CHR$(200)&CHR$(LEN(F$))&F$&CHR$(0):: FN=FN+1
180 M$=M$&CHR$(200)&CHR$(LEN(F$))&F$&CHR$(179):: IF X<11
 THEN 150
190 IF M$="" THEN 210
200 PRINT #2:CHR$(INT(LN/256))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&SEG$(M$,1,LEN(M$)-1)&CHR$(0):: LN=LN+1 :: M$=
"" :: X=0 :: IF LEN(F$)<>0 THEN 150
210 PRINT #2:CHR$(INT(LN/256))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&CHR$(200)&CHR$(3)&"END"&CHR$(0)
220 PRINT #2:CHR$(255)&CHR$(255):: CLOSE #1 :: CLOSE #2
230 DISPLAY AT(8,1)ERASE ALL :"Enter -":::" NEW":::" MER
GE DSK1.CAT":::" DELETE "" DSK1.CAT"":::" SAVE DSK1.LOA
D"
240 SUB TITLE(S,T$)
250 CALL SCREEN(S):: L=LEN(T$):: CALL MAGNIFY(2)
260 FOR J=1 TO L :: CALL SPRITE(#J,ASC(SEG$(T$,J,1)),J+1
-(J+1=S)+(J+1=S+13)+(J>14)*13,J*(170/L),10+J*(200/L)):
NEXT J
270 SUBEND
```

Next, enter MERGE DSK1.CAT/O and that "object code" will pop into place right after line 9. If you list it, it will look like a blown file, because most of the token codes are unprintable, but don't worry. Save the program as CATWRITER.

When you run the program, it will open an output MERGE format file called CAT and write those merged lines from CAT/O in MERGE format. Then it will open the disk you are cataloging, read the directory sector, and ask you for a disk name with the existing diskname as default. You can select any disk name you want to title the menu screen, up to 28 characters long. Line 110 computes the position to center the title, and lines 120-140 write to the CAT file a tokenized line 11 (overwriting that REM line) to display your title at the top of the screen.

Line 150 reads each filename from the disk directory, skipping over anything that is not a program (no one yet has been able to tell me how to distinguish an assembly image "program"!). For each filename, it will ask you for a complete program name. If you don't want a program on the menu (such as an XB program that is run from another program, or an image file), just press Enter. Otherwise the program name you select will be printed as DATA by line 170, in tokenized format in lines starting with 1100 (note the FN=1100 in line 100) and incremented by 1. Lines 180-200 assemble the filenames into DATA lines of up to ten names, and tokenize them in lines beginning with 1000.

When the last filename has been read, line 210 prints one last DATA item "END" to signal line 13 to stop reading, and then prints the double-255 end-of-file. Then you are given instructions to clear memory with NEW, merge in the CAT file, delete it because you don't need it any more, and save it back as LOAD.

When you list the LOAD program, you will find the original CAT/S restored in lines 10-19 and 1000, the

line to display the title in line 11, the filenames in DATA lines starting with 1000 and the program names in DATA lines starting at 1100.

When you run the program, it will display the disk name, and read the filenames into an array. Then it will display the program names, numbered, on as many screens as necessary, and ask you to select a program by number. The corresponding filename by number is selected from the array, and lines 19-20 rewrite line 10000 to RUN that filename. List the LOAD program after you have used it to load something, and you will see that it has changed.

That algorithm in lines 19-20 was published in one of the earliest 99'ER magazines, in a letter by A. Kludge. It has been the basis for every XBasic menu loader, and has saved us uncounted thousands of hours. The author had asked me not to reveal his identity, but I think I can now tell you that "A. Kludge" was really the late Dr. Stefan-Romano, who passed away recently at the age of 57. He was a brilliant man who did much for the TI world, at first as editor of the IUG library, and then through the Amnion library and Amnion Helpline. He was of great help to me on several occasions.

Some of you may have obtained from me a copy of CATWRITER which wrote GOSUB 21 in line 12, and CALL LOADs in lines 21-25 to change the cursor to my Tigercub emblem. If you have begun to have problems with the resulting LOAD program or with my previous Tigercub Menuloader which used the same CALL LOADs, I have finally found out the cause. When my Horizon RamDisk is on, any program containing those CALL LOADs will lock up the second time it is run!

PROGRAMS THAT WRITE PROGRAMS
Part 5
by Jim Peterson

In addition to writing programs in MERGE format, the same techniques can be used to analyze or modify programs which have been SAVEd in MERGE format. The D/V 163 file editor in Part 2 of this series was an example.

Here is a simple program to remove REM statements -

```
100 DISPLAY AT(3,3)ERASE ALL :"REM REMOVER": : :"Program must be SAVEd in":"MERGE format by":"SAVE DSK(filename),MERGE"
110 DISPLAY AT(12,1):"FILENAME? DSK" :: ACCEPT AT(12,14):F$ :: DISPLAY AT(14,1):"NEW FILENAME? DSK" :: ACCEPT AT(14,18):NF$
120 OPEN #1:"DSK"&F$,VARIABLE 163,INPUT :: OPEN #2:"DSK"&NF$,VARIABLE 163,OUTPUT
130 LINPUT #1:M$ :: A=POS(M$,CHR$(131),1):: B=POS(M$,CHR$(154),1):: A=MAX(A,B):: IF A=3 THEN 150 :: IF A=0 THEN PRINT #2:M$ :: GOTO 150
140 PRINT #2:SEG$(M$,1,A-1)&CHR$(0)
150 IF EOF(1)<>1 THEN 130 :: CLOSE #1 :: PRINT #2:CHR$(255)&CHR$(255):: CLOSE #2
```

The REM statement will begin with either a !, which is CHR$(131), or REM which is CHR$(154). So, line 130 reads in the lines one at a time. A finds the position in the line of ! and B finds the position of REM; one or the other, or both, will not be present and will equal 0. Then MAX finds the larger of A and B, which will be whichever one is present, or 0 if neither.

If ! or REM is in the 3rd position, immediately after the 2-byte line number, we want to delete the line entirely, so we do not reprint it. If A=0 then neither ! nor REM is present, so we reprint the entire line in the new file.

Otherwise, the REM statement is obviously a tail remark, so we reprint to the new file the segment of it starting with the first character and consisting of the number of characters one less than the position of the ! or REM. And, since we have lopped off the end of the line, we do not forget to replace the end-of-line marker CHR$(0).

If we have not reached the end of the file, we go back for the next line. Otherwise, we close the old file, but we remember to add the end-of-file marker to the new file before we close that too.

CORCOMP REPAIR FACILITY
IDT, Inc.
2211 E. Winston Road
Suite G
Anaheim, CA 92806

Phone 714-635-1815
Call 714-965-4460 for return material authorization

NEW

MYARC REPAIR FACILITY
Cecure Electronics
PO Box 132
Muskego, WI 53150

Phone 414-679-4343

TRANSFERRING FILES TI TO PC

by Philip Harris
Ottawa U.G.
December 1992

In case you're not sure how I passed TI Writer files to the PC, then let me explain. You will need the following hard/software: a null modem connector (approx. $8 at most computer stores), a female/female gender changer ($8), one or two modem cables, a modem program for both the PC and TI, TI Writer (Funnelweb), and the DF128/DV80 program (available from the Sysop on the Ottawa BBS).

Firstly, I download all the newsletter contributions from our BBS. After de-archiving any archived files, I then load up TI Writer.

Next, I select the first file for the newsletter and load the file as normal into TI Writer. Then merge in the next file by typing LF for Load File, and then type E DSK1.filename. This loads the file called "filename" from DSK1 and places it in the current document after the last line.

After all files are merged in, I can then do a rough deletion of unwanted formatting codes, then PF (Print File) to a file instead of a printer from the command line. The commands are: PF <enter>, C DSK1.Document <enter>. The "C" tells TI Writer to print the file stripping it of any TI Writer format characters and storing it in a "plain text" format.

The next step is to exit TI Writer and run the program DF128/DV80 and convert the DSK1.Document from a DV80 file to a DF128 file (ASCII standard).

Once the file conversion is complete, hook the two computers' serial ports together using the above equipment. Actually, you could do it DURING or even before the conversion.

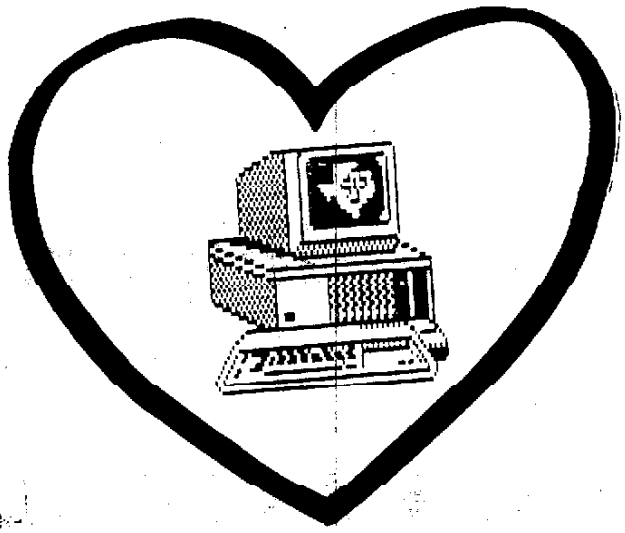
Lastly, run a modem program on each computer, enter terminal mode and set the baud rate at 9600. You should now be connected! You can test this by typing a few letters on the TI, and they should appear on the PC's screen. To transfer the files, simply select "upload" on the TI, Xmodem protocol, and type in the name of the file to send (e.g. in Telco: DSK1.DOCUMENT). Before pressing enter on the TI to send, select "download" on the PC, Xmodem, plus the file name (including path if needed) and press enter on both machines. The files should "scream" between machines. After the transfer, load your PC's word processing package (e.g. WordPerfect) and load in the ASCII text file. It's that simple!

ADDENDUM by Mark Schafer: I have successfully transported files between a TI and a PC, and the only connectors I used were a TI modem cable and a null modem cable (as opposed to a null modem connector). I didn't use a gender changer. Maybe you only need one if you don't have a null modem cable. I have no idea how easy a null modem cable is to come by.

If you think his second step reads a little awkwardly, it's because I changed it. He was going through a little more trouble than he needed to, so I simplified it. I couldn't say "I then merge..." because it isn't the way the author said he did it.

And speaking of unneeded trouble, I may be off the mark since I don't know a whole lot about telecommunications, but isn't it true that in some cases it would be easier just to download the file to a PC in the first place?

Also, I believe the above steps could be reversed if you want to go the other way. If the program DF128/DV80 that he referred to cannot convert the other way, there are programs that can. The conversion could also be done at the PC before sending.



ENGINEERING NEWS

Adapting the face of a clock *into the placement of 12 function keys on the MIKey keyboard reduces visual dependency. The design may also reduce other forms of CTDs as users maintain a relatively straight hand-wrist angle.*

Rethinking the keyboard. As the most common interface in the computer age, the traditional keyboard is also a prime cause of CTDs. The Mnemonic Ikon Keyboard (MIKey) combats tendonitis and carpal tunnel syndrome, says its designer, Dr. Allan H. Grant, Chevy Chase, MD. MIKey features a shallow V-Shape and a Center-Peak that permits the user to function with a relatively straight hand-wrist angle—a more comfortable, less fatiguing work position, says Grant.

MIKey's 12-function keys are aligned in a circular pattern, simulating the 12 hours on the face of a clock. The clockface, universally recognized, can function as a mnemonic icon to lessen visual dependence on the keyboard.

"Current computer configurations are based on designs that have prevailed far past the point at which significant changes should have been initiated," says Grant. "Emphasis has favored the hardware itself, rather than harmonic interaction of man and machine."

LONG ISLAND SOUND
The Newsletter of the
LONG ISLAND SQer's U. G.

EDITOR: FRANK BUBENIK,JR

TONY IANNACONE.PRESIDENT
JERRY STUCKLER.VICE PRES
FRANK BUBENIK..SECRETARY
BOB LAWSON.....TREASURER

MEETINGS HELD ON SECOND
FRIDAY EACH MONTH AT
TONY'S HOME
542 SOUTH BROADWAY
LINDENHURST,NY

PHONE: (516) 938-1095
BBS NO:(516) 661-3643

> > MAILING ADDRESS < <
93 MYERS AVE
HICKSVILLE, NY
11801-2424

FIRST CLASS MAIL

DALLAS TI GROUP LISA SHAFFER
P.O. BOX 29863
DALLAS TX
75229 USA
9305 <-LAST NL RECEIVED